

SPICE
Project

Spice User Manual

Draft 10

Copyright © 2009 Red Hat, Inc.
Licensed under a Creative Commons Attribution-Share Alike 3.0 United States License (see http://creativecommons.org/licenses/by-sa/3.0/us/legalcode).

Table of Contents

1 Introduction....3
1.1 Spice and Spice-related Components....3
1.1.1 Spice Server....3
1.1.1 Spice Client....3
1.1.2 QXL Device and Drivers....3
1.1.3 VDI Port Device....3
1.1.4 Spice Agent....3
1.2 Features....4
1.2.1 Multiple Channels....4
1.2.2 Image Compression....4
1.2.3 Video Compression....4
1.2.4 Mouse Modes....4
1.2.5 Other Features....4
2 Installation....5
2.1 Prerequisites....5
2.1.1 qpixman....5
2.1.2 qcairo....5
2.1.3 celt_0_5_1....5
2.1.4 ffmpeg....6
2.1.5 log4cpp....6
2.2 Windows prerequisites....7
2.3 Obtaining the source....7
2.4 Building from source....7
2.4.1 libspice and Linux client....7
2.4.2 qemu/kvm....8
2.4.3 Windows Client....8
3 Running spice....9
3.1 Preparations....9
3.2 Examples....10
4 Spice server....10
4.1 Qemu spice options....10
4.1.1 Creating SSL files....12
4.2 Qemu monitor commands....12
4.2.1 Control....12
4.2.2 Information....13
5 Spice client....13
5.1 Command options....13
5.2 Control hot keys....14
6 Guest....15
6.1 Windows....15
6.1.1 Drivers Install....15
6.1.2 Agent Install....15
7 Logging and debugging....16
8 References....16

1 Introduction

Spice is an open remote computing solution, providing client access to remote displays and devices (e.g. keyboard, mouse, audio). Spice provides a desktop like user experience, while trying to offload most of the intensive CPU and GPU tasks to the client.

The basic building blocks of Spice are:

1. Spice protocol – see [1]
2. Spice server
3. Spice client

The following sections provide basic information on Spice components and features, obtaining, building, installing and using Spice.

1.1 Spice and Spice-related Components

1.1.1 Spice Server

Spice server is implemented in libspice, a VDI pluggable library. Virtual Device Interface (VDI) defines a set of interfaces that provide a standard way to publish virtual devices (e.g. display device, keyboard, mouse) and enables different Spice components to interact with those devices. For more information, refer to [2]. On one side, the server communicates with the remote client using the Spice protocol and on the other side, it interacts with the VDI host application (e.g QEMU).

1.1.1 Spice Client

Spice cross-platform (Linux & Windows) client is the interface for the end user.

1.1.2 QXL Device and Drivers

Spice server supports QXL VDI interface. When libspice is used with QEMU, a specific QEMU QXL PCI device can be used for improving remote display performance and enhancing the graphic capabilities of the guest graphic system. QXL device requires guest QXL drivers for full functionality. However, standard VGA is supported when no driver exists.

1.1.3 VDI Port Device

Spice protocol supports a communication channel between the client and the agent on the server side. When using QEMU, Spice agent resides on the guest. VDI port is a QEMU PCI device used for communication with the agent.

1.1.4 Spice Agent

The Spice agent is an optional component for enhancing user experience and performing guest-oriented management tasks. For example, the agent injects mouse position and state to the guest when using client mouse mode. In addition, it is used for configuration of the guest display settings.

1.2 Features

1.2.1 Multiple Channels

The server and client communicate via channels. Each channel is dedicated to a specific type of data. The available channels are:

- **Main** - control and configuration
- **Display** - graphic commands, images and video streams
- **Inputs** - keyboard and mouse inputs
- **Cursor** - pointer device position and cursor shape
- **Playback** - audio received from the server to be played by the client
- **Record** - audio capture on the client side

1.2.2 Image Compression

Spice offers several image compression algorithms, which can be chosen on server initiation and dynamically at run-time. Quic is a Spice proprietary image compression technology based on the SFALIC algorithm. The Lempel-Ziv (LZ) algorithm is another option. Both Quic and LZ are local algorithms encoding each image separately. Global LZ (GLZ) is another proprietary Spice technology that uses LZ with history based global dictionary. GLZ takes advantage of repeating patterns among images to shrink the traffic and save bandwidth, which is critical in a WAN environment. Spice also offers an automatic mode for compression selection per image, where the choice between LZ/GLZ and Quic is heuristically based on image properties. Conceptually, synthetic images are better compressed with LZ/GLZ and real images are better with Quic.

1.2.3 Video Compression

Spice uses loss-less compression for images sent to the client. However, video streams are handled differently. Spice server heuristically identifies video areas and sends them as a video stream coded using M-JPEG. This handling saves a lot of traffic, improving Spice performance, especially in a WAN environment. However, in some circumstances the heuristic behavior might cause low quality image (e.g. identifying updated text area as a video stream). Video streaming can be chosen on server initiation and dynamically at run-time.

1.2.4 Mouse Modes

Spice supports two mouse modes: server and client. The mode can be changed dynamically and is negotiated between the client and the server.

- **Server mouse** – When a user clicks inside the Spice client window, the client mouse is captured and set invisible. In this mode, the server controls the mouse position on display. However, it might be problematic on WAN or a loaded server, where mouse cursor might have some latency or non-responsiveness.
- **Client mouse** - Not captured and is used as the effective pointing device. For enabling client mouse, VDI host application must register an absolute pointing device (e.g. USB tablet in QEMU). This mode is appropriate for WAN or loaded server, since cursor has smooth motion and responsiveness. However, the cursor might loose synchronization (position and shape) for a while.

1.2.5 Other Features

- **Multiple Monitors** - any number of monitors is supported
- **Bidirectional Audio** - Spice supports audio playback and recording. Playback is compressed using the CELT algorithm
- **Lip-sync** - between video and audio. Available only when video streaming is enabled.
- **Migration** - switching channel connectivity for supporting server migration
- **Pixmap caching**

2 Installation

2.1 Prerequisites

The following list details the packages required for building Spice on Linux. It was tested on Fedora 11.

2.1.1 qpixman

Pixman is a generic library for manipulating pixel regions, containing low-level pixel manipulation routines and is used by Cairo. QPixman is a slightly modified version of Pixman.

For getting and installing qpixman:

Download qpixman from Spice site @ www.spice-space.org.

```
cd <qpixman_dir>
./autogen.sh -includedir=/usr/include --libdir=/usr/lib64
make
sudo make install
```

2.1.2 qcairo

Cairo is a vector graphics, device-independent library. qcairo is a slightly modified version of Cairo.

For getting and installing qcairo:

Download qcairo from Spice site @ www.spice-space.org.

```
cd <qcairo_dir>
./autogen.sh --disable-xlib --disable-ps --disable-pdf ▶
     --disable-svg --includedir=/usr/include/ --libdir=/usr/lib64
make
sudo make install
```

2.1.3 celt_0_5_1

The CELT codec is a compression algorithm for audio - transmitting music with high quality, imposing very little delay on the signal.

For getting and installing CELT:

Download celt-0.5.1.3.tar.gz from http://www.celt-codec.org/downloads/

```
cd <celt_dir>
./configure -includedir=/usr/include --libdir=/usr/lib64
make
sudo make install
```

2.1.4 ffmpeg

FFmpeg is a library for audio and video recording, conversion, and streaming. It includes libavcodec audio/video codec library.

In case your distribution does not include the library:

Download it from http://ffmpeg.org/

```
cd <ffmpeg_dir>
./configure --libdir=/usr/lib64 --incdir=/usr/include ▶
     --shlibdir=/usr/lib64 --disable-demuxers --disable-ffmpeg ▶
     --disable-ffserver --disable-ffplay --disable-bsfs ▶
     --disable-parsers --disable-devices --disable-protocols ▶
     --disable-muxers --disable-decoders --disable-encoders ▶
     --enable-encoder=mjpeg --enable-decoder=mjpeg ▶
     --disable-static --enable-shared --enable-gpl ▶
     --disable-stripping
make
sudo make install
```

2.1.5 log4cpp

Log4cpp is a library for flexible logging to files, syslog etc. It is modeled after the Log4j Java library.

In case your distribution does not include the library:

Download it from http://log4cpp.sourceforge.net/

```
cd <log4cpp_dir>

./configure -includedir=/usr/include --libdir=/usr/lib64
make
sudo make install
```

If the build fails, apply the following patch and try again:

```
diff -ru log4cpp-1.0.orig/src/BasicLayout.cpp log4cpp-1.0/src/BasicLayout.cpp
--- log4cpp-1.0.orig/src/BasicLayout.cpp        2006-09-30 02:03:20.000000000 -0400
+++ log4cpp-1.0/src/BasicLayout.cpp     2008-12-12 12:01:16.000000000 -0500
@@ -8,6 +8,8 @@
  */

 #include "PortabilityImpl.hh"
+#include <cstdlib>
+#include <memory>
 #include <log4cpp/BasicLayout.hh>
 #include <log4cpp/Priority.hh>
 #include <log4cpp/FactoryParams.hh>
diff -ru log4cpp-1.0.orig/src/PatternLayout.cpp log4cpp-1.0/src/PatternLayout.cpp
--- log4cpp-1.0.orig/src/PatternLayout.cpp      2007-08-28 03:54:12.000000000 -0400
+++ log4cpp-1.0/src/PatternLayout.cpp   2008-12-12 11:57:59.000000000 -0500
@@ -8,6 +8,7 @@

 #include "PortabilityImpl.hh"

+#include <cstdlib>
 #include <log4cpp/PatternLayout.hh>
 #include <log4cpp/Priority.hh>
 #include <log4cpp/NDC.hh>
```

2.2 Windows prerequisites

Building Windows components is optional, you can download the binaries from Spice site @ www.spice-space.org.

- Download `windows_libraries` archive from Spice site @ www.spice-space.org. and extract it.
- Set `SPICE_LIBS` environment variable to `<windows_libraries_dir>`
- Download and install openssl from: http://www.slproweb.com/products/Win32OpenSSL.html
- Add OpenSSL include and static lib directories (<OpenSSL_dir>\include and <OpenSSL_dir>\lib\VC\static) to VC default libraries. In Tools → Options → Projects and Solutions → VC++ Directories → Include Files and Libarry files.
- Note: the static libs and instructions are for MSVC++ 2005.

2.3 Obtaining the source

We assume both Spice and qemu/kvm are located under the same folder <spice_root>, therefore:

```
mkdir <spice_root>
```

Download spice and vdesktop tarball from Spice site @ www.spice-space.org and extract them into <spice_root> directory.

2.4 Building from source

2.4.1 libspice and Linux client

To build both libspice and Linux client:

```
cd <spice_root>/spice
./autogen.sh -includedir=/usr/include --libdir=/usr/lib64
make
sudo make install
```

Building them separately can be done from the libspice and red/x11 folders.

2.4.2 qemu/kvm

In case of an older kernel (version < 2.6.30)

Get kernel sources using the following git repository:
git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-2.6.30.y.git

```
cd <spice_root>/vdesktop/kernel
./configure
ln -s <linux-2.6.30 source dir> linux-2.6
make sync
make
```

In both cases:

```
cd <spice_root>/vdesktop
./configure --enable-spice
```

Verify that Spice, kvm, and QXL was successfully configured by the configure script

```
cd libkvm
make
cd ../qemu
make

ln -s x86_64-softmmu/qemu-system-x86_64 qemu
```

Download the bios tarball from Spice site @ www.spice-space.org and put the files in <spice_root>/vdesktop/qemu/pc-bios directory. In case you would like to build the bios from source:

```
cd <spice_root>/vdesktop/bios
make
ln -s BIOS-bochs-latest ../qemu/pc-bios/bios.bin
cd ../vgabios
make
ln -s VGABIOS-lgpl-latest.bin ../qemu/pc-bios/vgabios.bin
```

2.4.3 Windows Client

Building Windows components is optional, you can download the binaries from `Spice site @ www.spice-space.org`.

In case you would like to have your output in a different dirctory, set the REDC_BUILD_DIR environment variable to point to the alternative dirctory.

Open <spice_root>\spice\red\windows\redc.sln and build it.

3 Running spice

3.1 *Preparations*

In case you have built the kernel module in section 2.4.2, you have to load the kvm modules

```
insmod <spice_root>/vdesktop/kernel/x86/kvm.ko
insmod <spice_root>/vdesktop/kernel/x86/kvm-<intel/amd>.ko
```

Otherwise, `sudo lsmod | grep kvm`, to make sure kvm is loaded.

Creating an image:

- `cd <spice_root>/vdesktop/qemu`
- To create a raw image of 10GB:
  `qemu-img create /path/to/xp.img 10G`
- Insert the os install cd, and boot qemu from cd to install the os on the qemu image:
  `qemu -hda /path/to/xp.img -boot d -cdrom /dev/cdrom -m 512`
- After the install has finished, shut down the guest.
- For further details, refer to: http://www.nongnu.org/qemu/qemu-doc.html

Networking:

- `cd /etc/sysconfig/network-scripts/`
- In `ifcfg-eth0` , add `BRIDGE=sw0`
- Create a new `ifcfg-sw0` that has `TYPE=Bridge`
- `service network restart`

For example, `ifcfg-eth0` might look something like:

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
DHCPCLASS=
HWADDR=<MAC_ADDR>
BRIDGE=sw0
```

And `ifcfg-sw0:`

```
DEVICE=sw0
TYPE=Bridge
BOOTPROTO=dhcp
ONBOOT=yes
USERCTL=no
DELAY=0.1
STP=off
```

3.2 Examples

Basic invocation of qemu with spice enabled, looks something like:

```
qemu -L ./pc-bios -m 512 -hda /path/to/xp.img -boot c
-net nic,macaddr=<UNIQUE_MAC_ADDR>,model=rtl8139
-net tap,ifname=XP1,script=/path/to/lan-ifup
-monitor stdio -usbdevice tablet -soundhw ac97
-qxl 1 -spice port=5930,disable-ticketing
```

If you like to save the output to log file add:

```
| tee <log file name>
```

This will launch a VM with 512MB of RAM, booting from hd0 with the image xp.img. NIC with a unique MAC address and its corresponding TAP network interface are enabled. Unique address can be generated using the physical NIC address with some changes to the significant bits.

The lan-ifup script might be:

```
#!/bin/sh
switch=sw0
ifconfig $1 0.0.0.0 up
brctl addif ${switch} $1
```

Qemu monitor is redirected to stdio, a USB tablet device is attached and ac97 audio device is used. Single QXL device is enabled and the Spice server listens on port 5930 . Ticketing is disabled, so client connects with no password.

For additional qemu information, refer to http://www.qemu.org/qemu-doc.html

Running a client, connecting to server <server> on port <port>:

```
spicec -h <server> -p <port>
```

4 Spice server

4.1 Qemu spice options

To enable the qxl device(s) and spice, qemu is executed with the following options:

```
qemu [qemu_options] -qxl <num> -spice <spice_options>
```

The qxl option:

```
-qxl <num> [,ram=megs]
```

Set 'num' qxl display devices, each with RAM size of 'megs' MB (current default=64MB)

The spice option:

```
-spice <spice_options>
```

Basic options

`[port=<port>][,sport=<sport>][,host=<host>]`
Listen on interface address <host> (if given, otherwise any interface)
using port <port> and/or sport <sport> (at least one of them must be given)

Image, video & audio options

`ic=on|auto_glz|auto_lz|quic|glz|lz|off`
Set image compression (default=on=auto_glz)
`quic` is based on the SFALIC algorithm
`lz` is the Lempel-Ziv algorithm, `glz` uses `lz` with history based global dictionary
The `auto_[glz/lz]` modes choose between the `[glz/lz]` and `quic`,
based on the image properties

`sv=<on|off>`
Set video streams detection and (lossy) compression (default=on)

`renderer=<oglpbuf+oglpixmap+cairo>`
Select renderers. Multiple choice prioritized by order (default=cairo)

`playback-compression=<on|off>`
Set playback compression, using the CELT algorithm (default=on)

Security options

`disable-ticketing`
Enables client connection with no password.

`password=<password>`
Set ticket password, which must be used by a client for connection. The passwords never expires.

`sslkey=<key_file>`
SSL private key file

`sslcert=<cert_file>`
SSL certificate file (self signed or chain of certificates). The certificate is currently not verified by the client, and used only for sending the server public key to the client.

`sslcafile=<ca_file>`
SSL certificates file of the trusted CA (certificate authority) and CRL (certificate revocation list)

`ssldhfile=<dh_file>`
Symmetric Diffie-Hellman key file

`sslpassword=<pem_password>`
Password to open the private key file which is in PEM format

`sslciphersuite=<cipher_suite>`
Cipher suite to use, see http://www.openssl.org/docs/apps/ciphers.html

`secure-channels=[all|ch0+ch1+...]`
`unsecure-channels=[all|ch0+ch1+...]`
Force secured/unsecured connection on all/specific channels
Channels are: `main, display, inputs, cursor, playback` and `record`
By default, any channel allows both secured and unsecured connection, depending on the `port` and `sport` parameters.

Other options
`agent-mouse=<on|off>`
Define whether spice agent is used for client mouse mode (default=on)

4.1.1 Creating SSL files

Below are examples for manually creating SSL files for use by the spice server.
For more detail, refer to http://www.openssl.org/docs/apps/openssl.html

Create key file:
```
openssl genrsa -out key.pem 1024
```

Create self-signed certificate, which can be used as CA file as well:
```
openssl req -new -x509 -key key.pem -out cert.pem -days 1095
```

Create DH file:
```
openssl gendh -out dh.pem 1024
```

4.2 Qemu monitor commands

4.2.1 Control

`spice.set_image_compression <on|auto_glz|auto_lz|quic|glz|lz|off>`
refer to the `ic` spice option

`spice.set_streaming_video <on|off>`
Set video streams detection and compression

`spice.set_playback_compression <on|off>`
Set playback compression

`spice.set_ticket <password> [expiration=<seconds>`
`[,connected=keep|disconnect|fail]`
Set the spice connection ticket (one time password), with expiration and behavior when already connected. An empty password prevents any connection. A zero value for expiration means the password never expires.

`spice.set_ticket64`
Same as `set_ticket`, but using base64 for printable ASCII password.

`spice.disable_ticketing`
Entirely disables password, so any client is allowed to connect.

`spice.set_agent_mouse <on|off>`
Define whether guest agent is used for client mouse mode

`spice.reset_stat`
Reset spice statistics

`spice.ping_client [on [interval]|off]`
Ping spice client to measure round-trip. Single ping needs no parameter. Repeating ping is started with 'on' and sent every 'interval' (default is 1 second). Pinging is stopped with 'off'.

4.2.2 Information

`info spice.state`
Show current spice state, including client address (if connected), ticketing, image compression, video streaming and playback compression.

`info spice.ticket`
Show ticket expiration (if enabled)

`info spice.stat`
Show spice statistics

`info spice.rtt_client`
Show round trip time (minimum, maximum & average) to spice client after pinging it.

5 Spice client

5.1 Command options

`spicec -h <host> [-p <port>] [-s <secure-port>] [options]`

`-h, --host <host>`
Spice server address

`-p, --port <port>`
Spice server port

`-s, --secure-port <port>`
Spice server secure port

`--secure-channels <ch0,ch1...>`
Force secure connection on the specified channels
Channels are: `main, display, inputs, cursor, playback` and `record`
By default, any channel can be secured, depending on the `--secure-port` availability.

`--unsecure-channels <ch0,ch1...>`
Force unsecure connection on the specified channels
Channels are: `main, display, inputs, cursor, playback` and `record.`
By default, any channel can be unsecured, depending on the `--port` availability.

`-w, --password <password>`
Set a ticketing password (default is none)

`-f, --full-screen[=auto-conf]`
: Open in a full screen mode. Optional automatic configuration of the remote display settings according to the client display settings. A guest agent must be installed and started for this option.

`--canvas-type <type1, type2...>`
: Select the available renderer. The order determines precedence

  For Linux client : only “cairo” canvas type is available.

  For Windows client : available types are “gdi” and “cairo”. The default is “gdi”.

`--enable-channels <ch0,ch1...>`
: Enable the specified channels. Use “all” for enabling all possible channels. Use the following names for enabling only the selected channels: “display”, “inputs”, “cursor”, “playback” and “record”. By default all channels are enabled.

`--disable-channels <ch0,ch1...>`
: Disable the specified channels. Use “all” for disabling all possible channels. Use the following names for enabling only the selected channels: “display”, “inputs”, “cursor”, “playback” and “record.” By default all channels are enabled.

`--help`
: Show command help

5.2 Control hot keys

`Shift + F11`
: Toggle full-screen / window mode

`Shift + F12`
: Release the cursor if captured in window

On debug mode, these are available as well:

`Shift + F5`
: Connect to the server

`Shift + F6`
: Disconnect from the server

6 Guest

6.1 Windows

To get Windows binaries: Download `windows_binaries` archive from Spice site @ www.spice-space.org.

6.1.1 Drivers Install

For installing each of the QXL device drivers (first time or after uninstall):

- Put the installation information (qxl.inf), miniport driver (qxl.sys) and display driver (qxldd.dll) files in a directory visible by the guest.
- When Windows finds new hardware, choose not to connect to Windows Update.
- Choose 'Install from a list or specific location'.
- Choose 'Search for best driver in these locations' and set the drivers directory as the search path.

To remove the currently installed QXL drivers, in case of problem in reinstalling drivers:
Restart after completion of the following. If one of the files cannot be deleted, try again after restart, before installing new drivers.

- Right-click 'My Computer' and choose 'Manage'.
- In the 'Computer Management' window, expand the 'Display adapters'.
- Uninstall all the 'Red Hat QXL GPU' device drivers.
- Assuming <windir> is the Windows directory (e.g. C:\Windows), in <windir>\inf, find and delete all the oemX.inf files which include installation information for the qxl driver, and their respective oemX.PNF files.
- In <windir>\system32, delete qxldd.dll.
- In <windir>\system32\drivers, delete qxl.sys.

vdiport driver install and removal is performed similarly to the QXL driver, but this time using the vdi_port.inf and vdi_port.sys files instead.

6.1.2 Agent Install

The Windows agent is consisting of a system service (`vdservice.exe`) and a user process (`vdagent.exe`).

Note:

- The service currently supports only Windows XP.
- Install the service only after installing vdiport driver.

To install and start the service:

- Put `vdservice.exe` and `vdagent.exe` in the same directory.
- `vdservice install`
- `net start vdservice`

To stop and remove the currently installed service:

- `net stop vdservice`
- `vdservice uninstall`

7 Logging and debugging

Spice server logs to stdout
Spice client log is /tmp/spicec.log in Linux. In Windows, it is %temp%\spicec.log.

8 References

[1] Spice remote computing protocol definition

[2] Spice VD Interfaces documentation